

MÉTODOS NUMÉRICOS

SHORTEST PATH PROBLEM

El problema del camino más corto

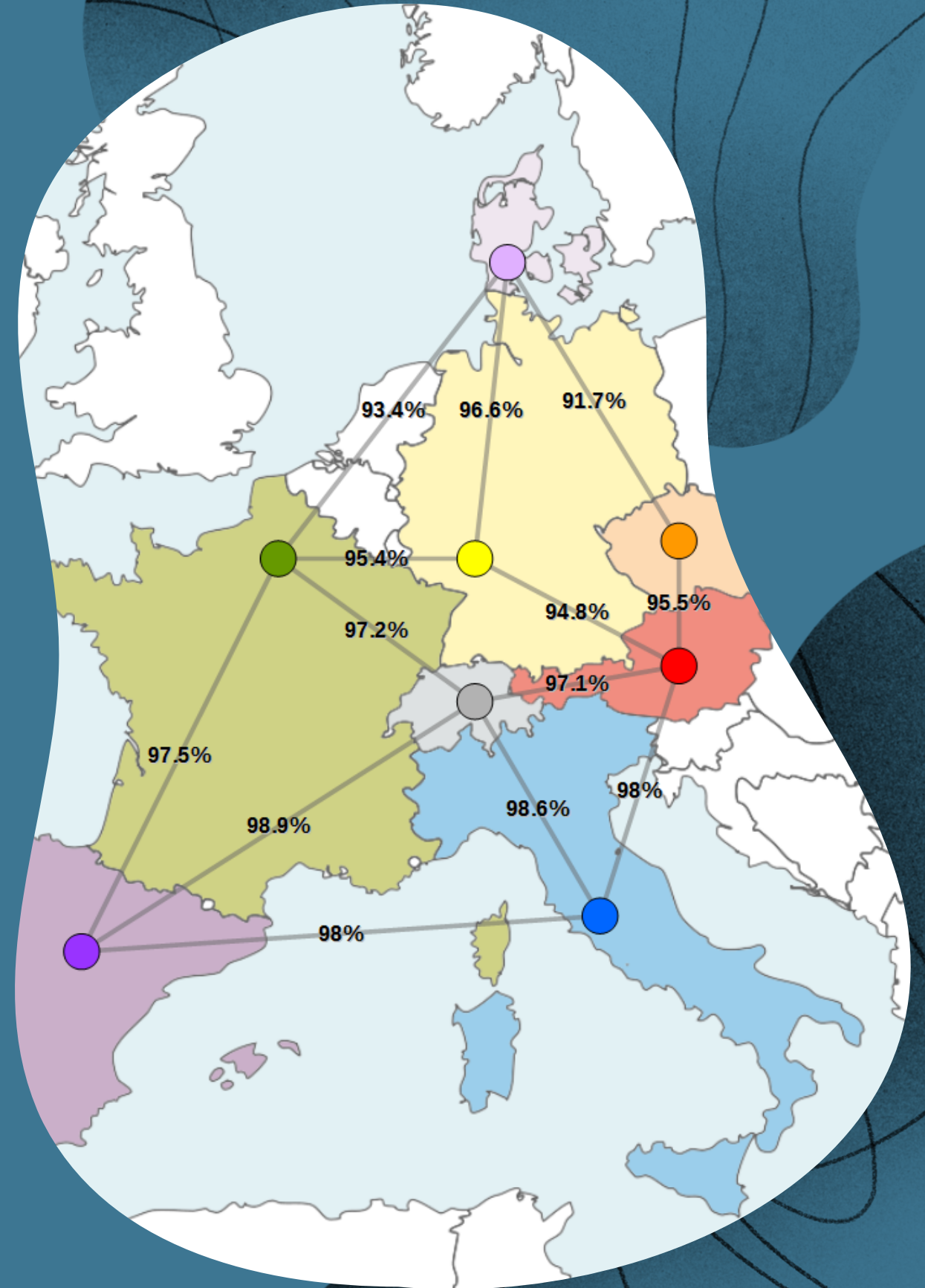
Elisa Samayoa | 20710

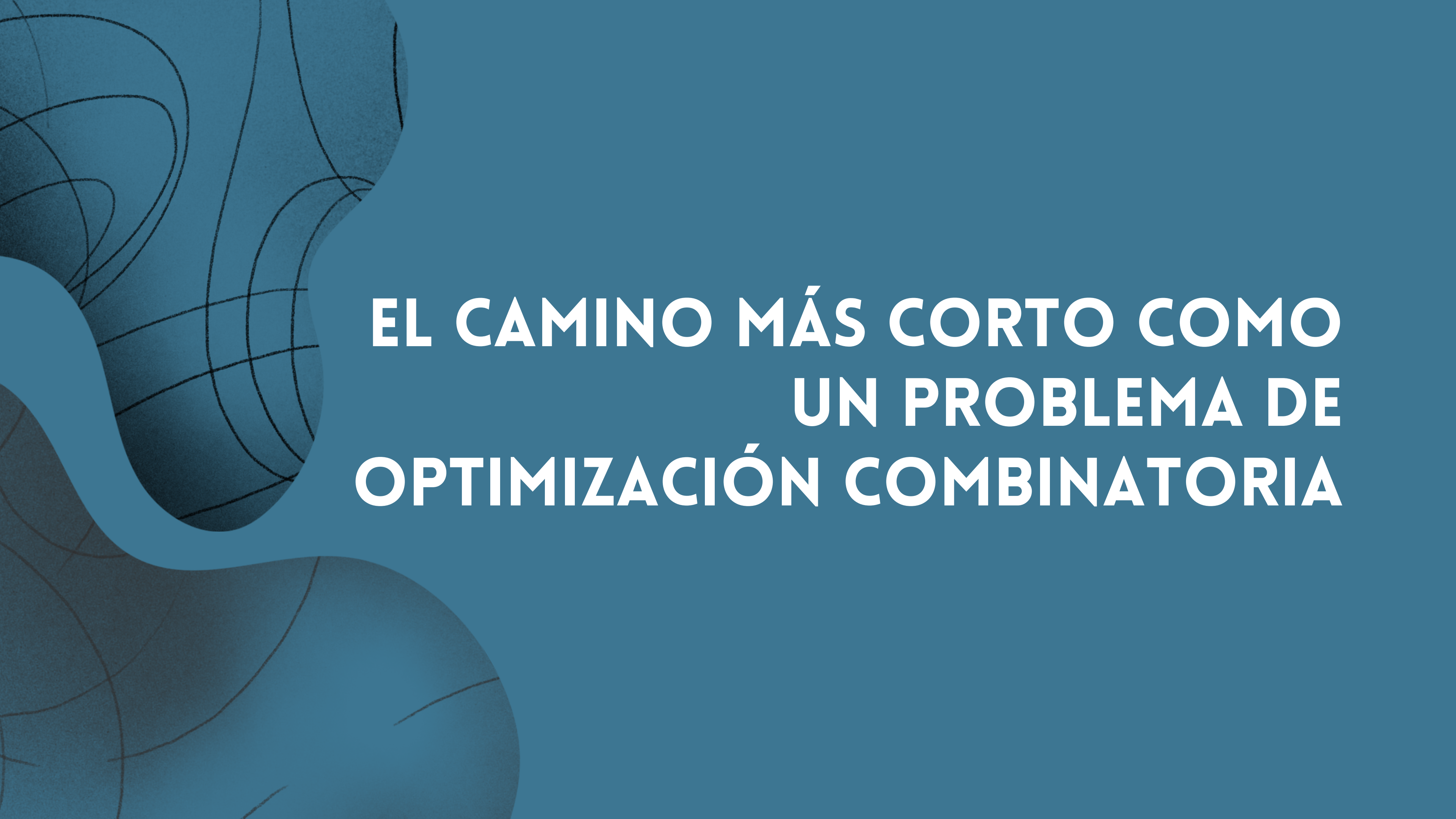
Julio Avila | 20333

INTRODUCCIÓN

El problema del camino más corto se basa en **encontrar un camino entre dos vértices u , v** que, al sumar los pesos de sus aristas, se obtenga el **valor mínimo**. Este problema puede ser usado tanto en gráficos dirigidos como no dirigidos.

En esta presentación analizaremos distintos algoritmos para abordar el problema del camino más corto y encontrar el más óptimo.





EL CAMINO MÁS CORTO COMO UN PROBLEMA DE OPTIMIZACIÓN COMBINATORIA

¿PROBLEMAS DE OPTIMIZACIÓN COMBINATORIA?

- Cantidad **finita** o numerable de soluciones
- Resuelven **problemas** que se consideran **difíciles** al explorar las posibles soluciones
- Logran **reducir el tamaño efectivo del espacio** de búsqueda y **explorarlo** de manera eficiente.
- Los procesos de resolución **no garantizan una solución óptima**, pero la aproximación al óptimo suele ser suficiente

**LA CALIDAD DE
LA SOLUCIÓN
NOS AFECTA
DIRECTAMENTE**

IMPORTANCIA EN DIVERSOS CAMPOS

Este problema es esencial en la optimización de procesos y recursos en una variedad de campos, lo que resulta en una **mayor eficiencia y ahorro de recursos**. Su aplicación mejora la **toma de decisiones** y contribuye significativamente al **rendimiento**.





ALGORITMO 1: DIJKSTRA

ALGORITMO DIJKSTRA

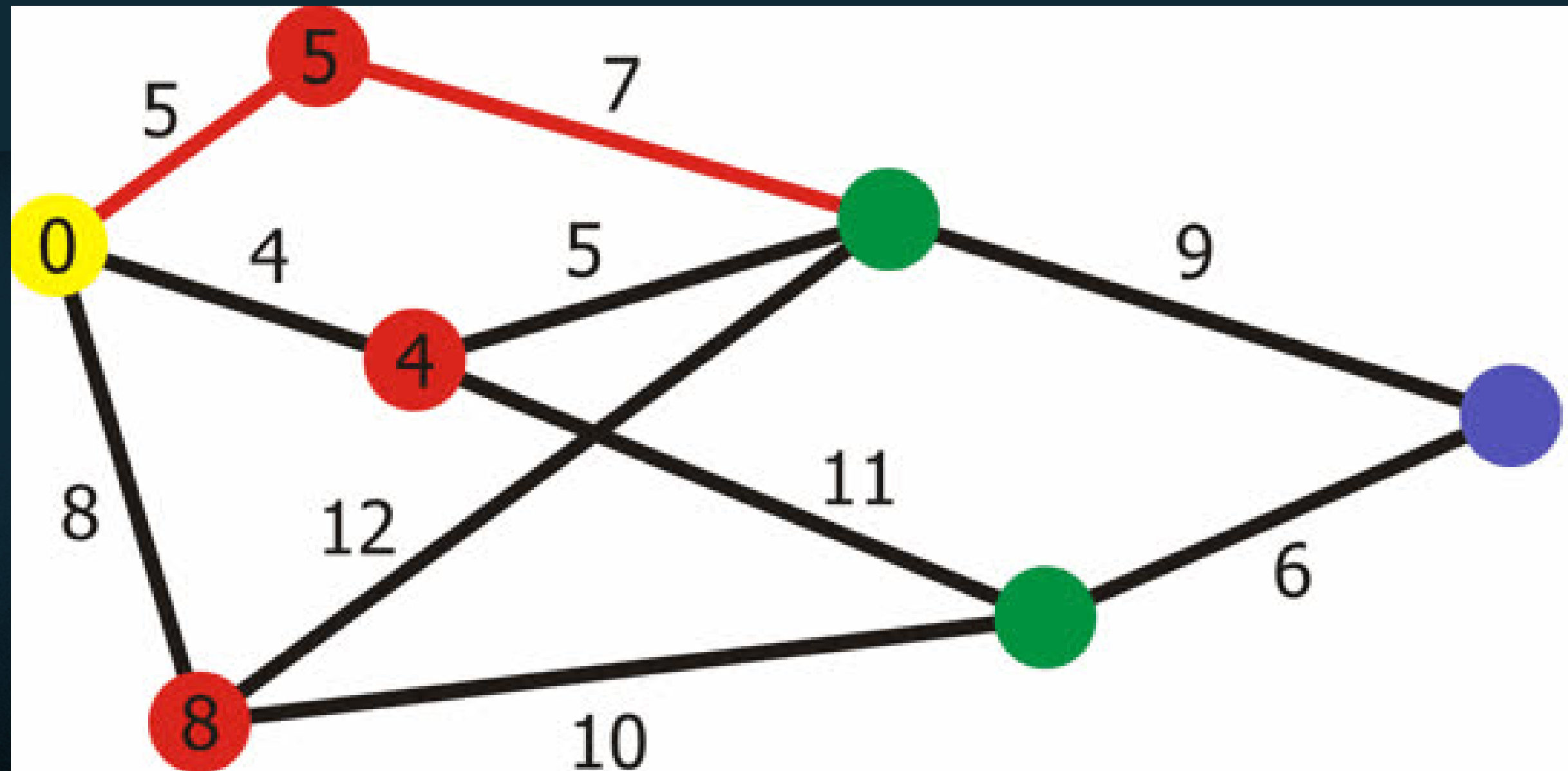
- Diseñado por el holandés **Edsger Wybe Dijkstra en 1959**.
- Es un algoritmo eficiente de complejidad $O(n^2)$, donde “n” es el número de vértices)
- Los pesos de las aristas deben ser no negativos



Se basa en el **principio de optimización**, donde busca encontrar el **camino más corto** entre dos vértices en un grafo con pesos asociados a sus arcos.

Dijkstra encuentra de manera progresiva los caminos de costo mínimo desde un vértice inicial hacia todos los demás vértices del grafo. Para lograrlo, selecciona en cada paso el vértice más cercano al vértice de origen cuya distancia aún no ha sido determinada. Al marcar este vértice como conocido, se establece el camino más corto desde el origen a este punto. Repite este proceso hasta que todos los vértices estén marcados,

EJEMPLO DEL ALGORITMO DIJKSTRA



**CÓDIGO
IMPLEMENTADO**



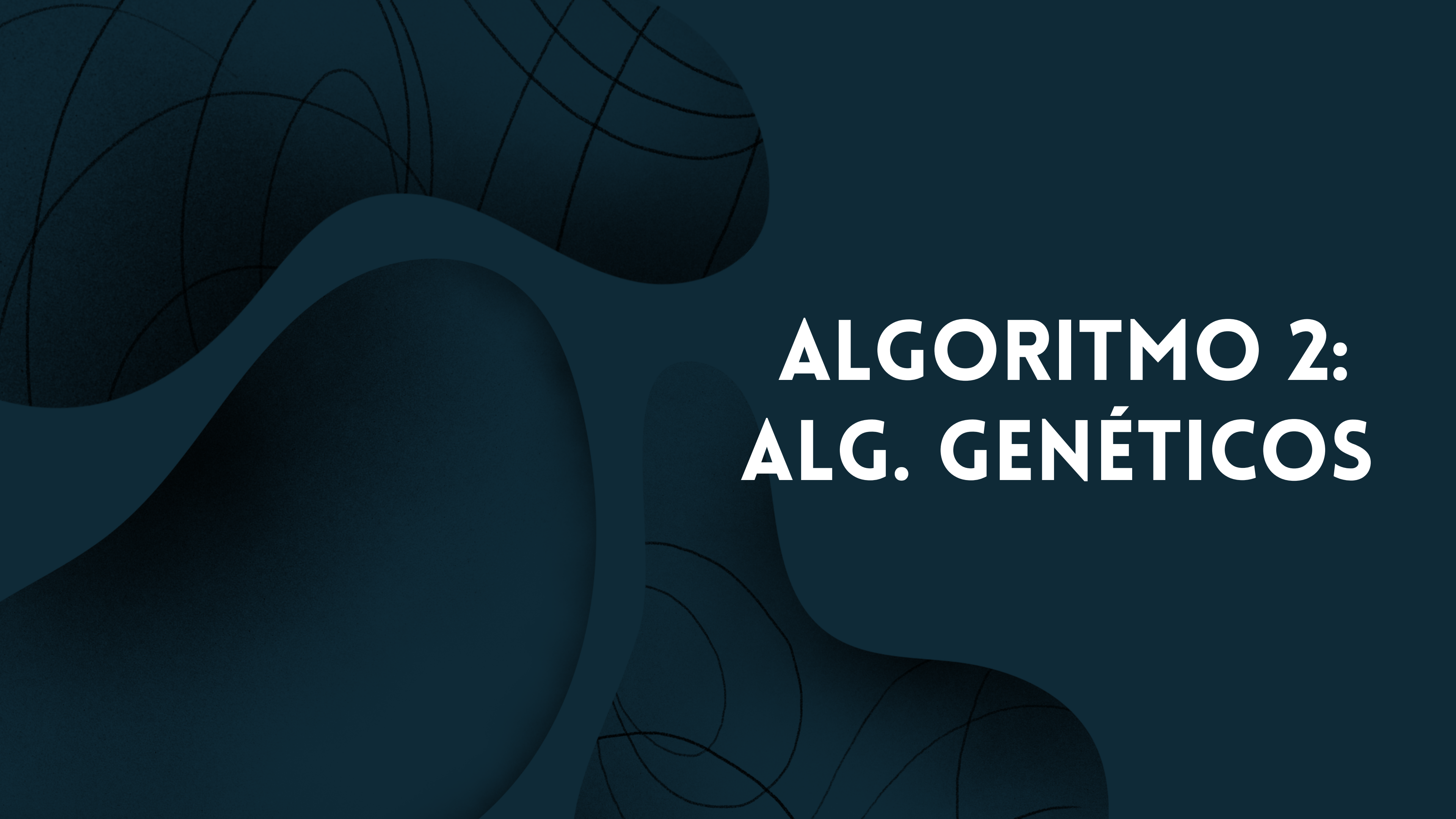
VENTAJAS

- Alta eficiencia
- Simplicidad
- Capacidad de devolver una solución óptima



LIMITACIONES

- Incapacidad de trabajar con pesos negativos
- Complejidad alta en grafos grandes
- Rendimiento inadecuado en escenarios específicos (ejemplo: grafos con ciclos negativos)



ALGORITMO 2: ALG. GENÉTICOS

ALGORITMOS GENÉTICOS

- Se popularizó con el trabajo de John Henry Holland
- No se adecuan bien a la complejidad, con un gran número de elementos expuestos a la mutación, hay un un aumento exponencial en el tamaño del espacio de búsqueda.

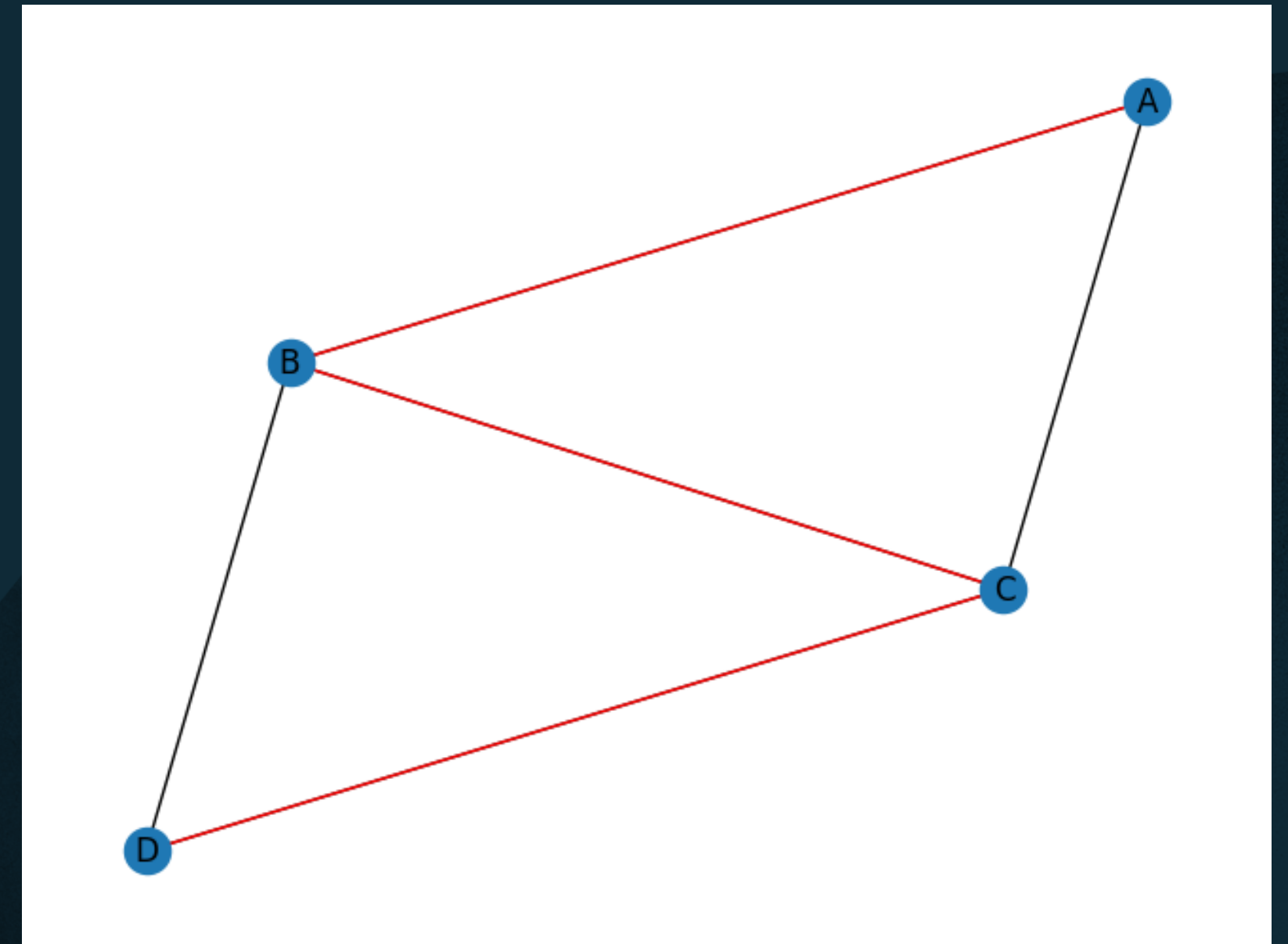


Se inspira en la **teoría de la evolución de Darwin**.

Es una solución de optimización basada en la selección natural y cambia varias veces una población de soluciones individuales. En cada paso, el algoritmo genético elige individuos al azar de la población actual para ser padres y los utiliza para reproducir los hijos de la siguiente generación. A lo largo de generaciones sucesivas, la población "evoluciona" hacia una solución óptima.

EJEMPLO DEL ALGORITMO GENÉTICO

- Camino más corto de A a D: ['A', 'B', 'C', 'D']
- Distancia del camino más corto: 6
- Población: 100
- Generaciones máximas: 50



**CÓDIGO
IMPLEMENTADO**



VENTAJAS

- Operan de forma simultánea con varias soluciones
- Cuando se usan para problemas de optimización (maximizar una función objetivo)
- resultan menos afectados por los máximos locales que otras soluciones
- Usan operadores probabilísticos, en vez de los típicos operadores determinísticos de otras técnicas



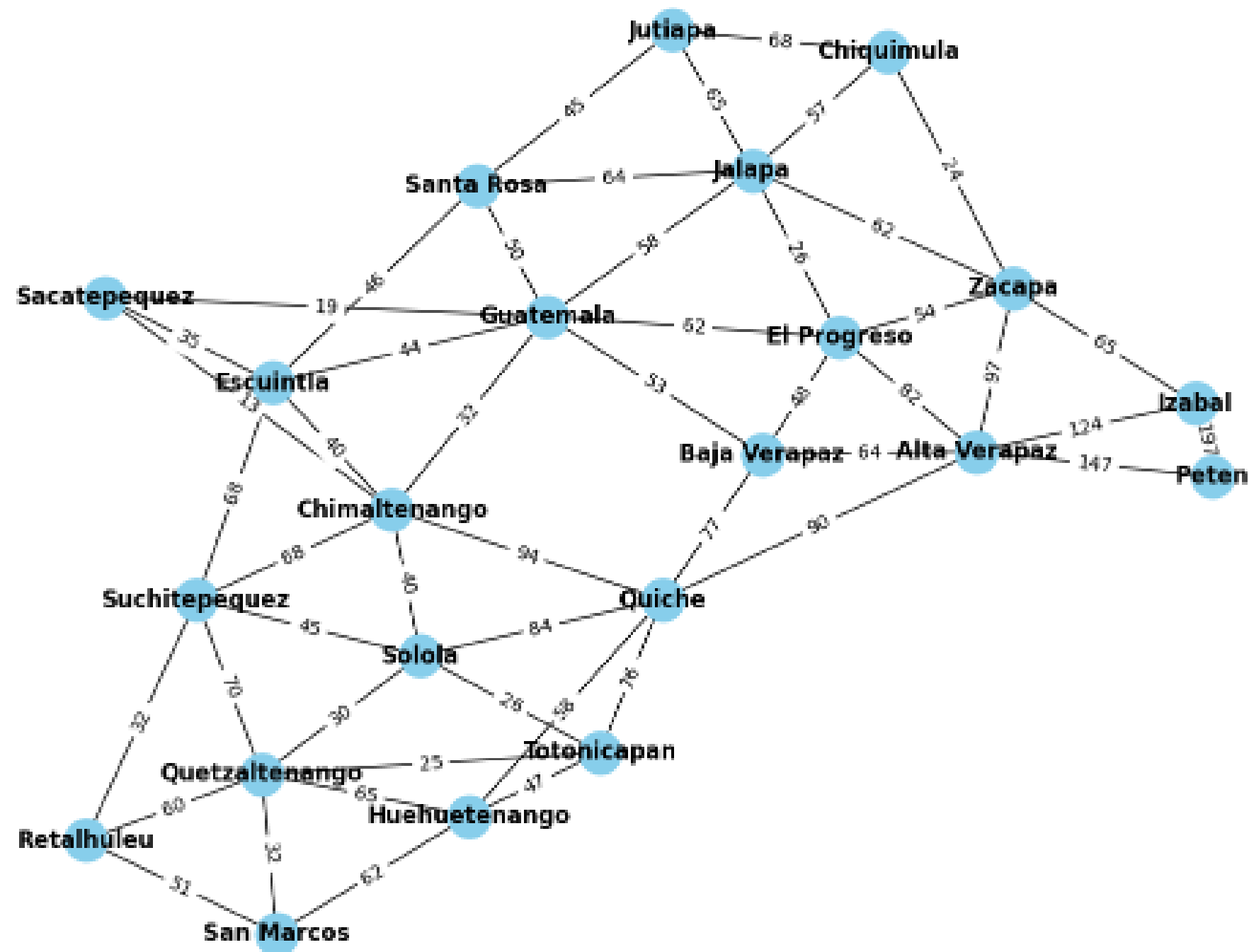
LIMITACIONES

- Costoso en la función de evaluación (tiempo, recursos)
- Podría no converger o converger antes con resultados insatisfactorios (óptimo local o punto arbitrario)
- No tiene buena escalabilidad con la complejidad
- No recomendable utilizarlos para problemas con soluciones simples

COMPARACIÓN DE SOLUCIONES



Grafo de Todas las Conexiones entre Departamentos de Guatemala



ENCONTRANDO LA RUTA MÁS CORTA ENTRE LOS DEPARTAMENTOS DE GUATEMALA

Se generó un grafo en el que cada nodo es un departamento de Guatemala y las aristas tienen un peso de la distancia en línea recta entre las cabeceras departamentales; se buscará la ruta más corta entre dos departamentos mediante los dos algoritmos y se compararán las soluciones obtenidas.

COMPARACIÓN DE SOLUCIONES

Ruta más corta de Retalhuleu a Petén

01. Dijkstra

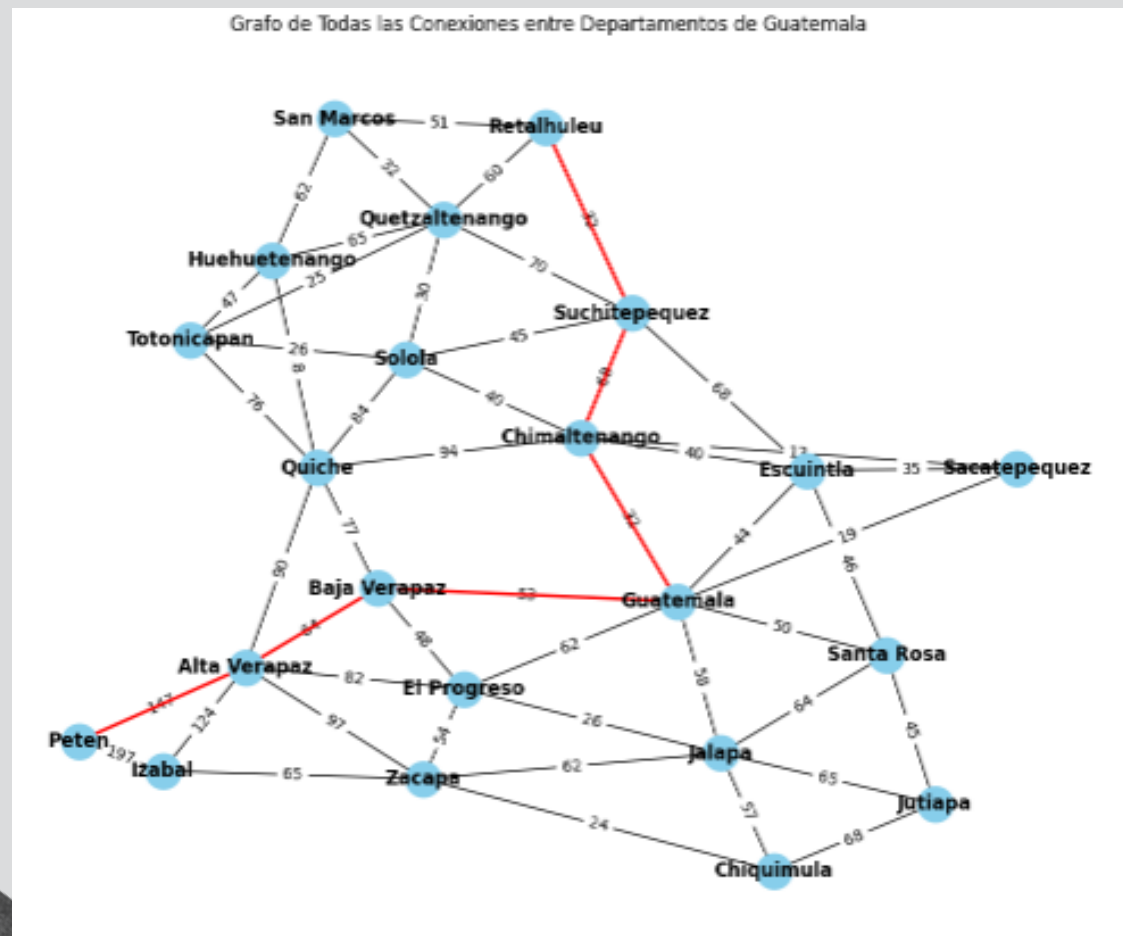
- El camino más corto entre Retalhuleu y Peten es:
['Retalhuleu', 'Suchitepequez', 'Chimaltenango', 'Guatemala', 'Baja Verapaz', 'Alta Verapaz', 'Peten']
- La distancia es: 396 km
- Tiempo de ejecución:
0.0002231597900390625 segundos

02. Alg. Genético

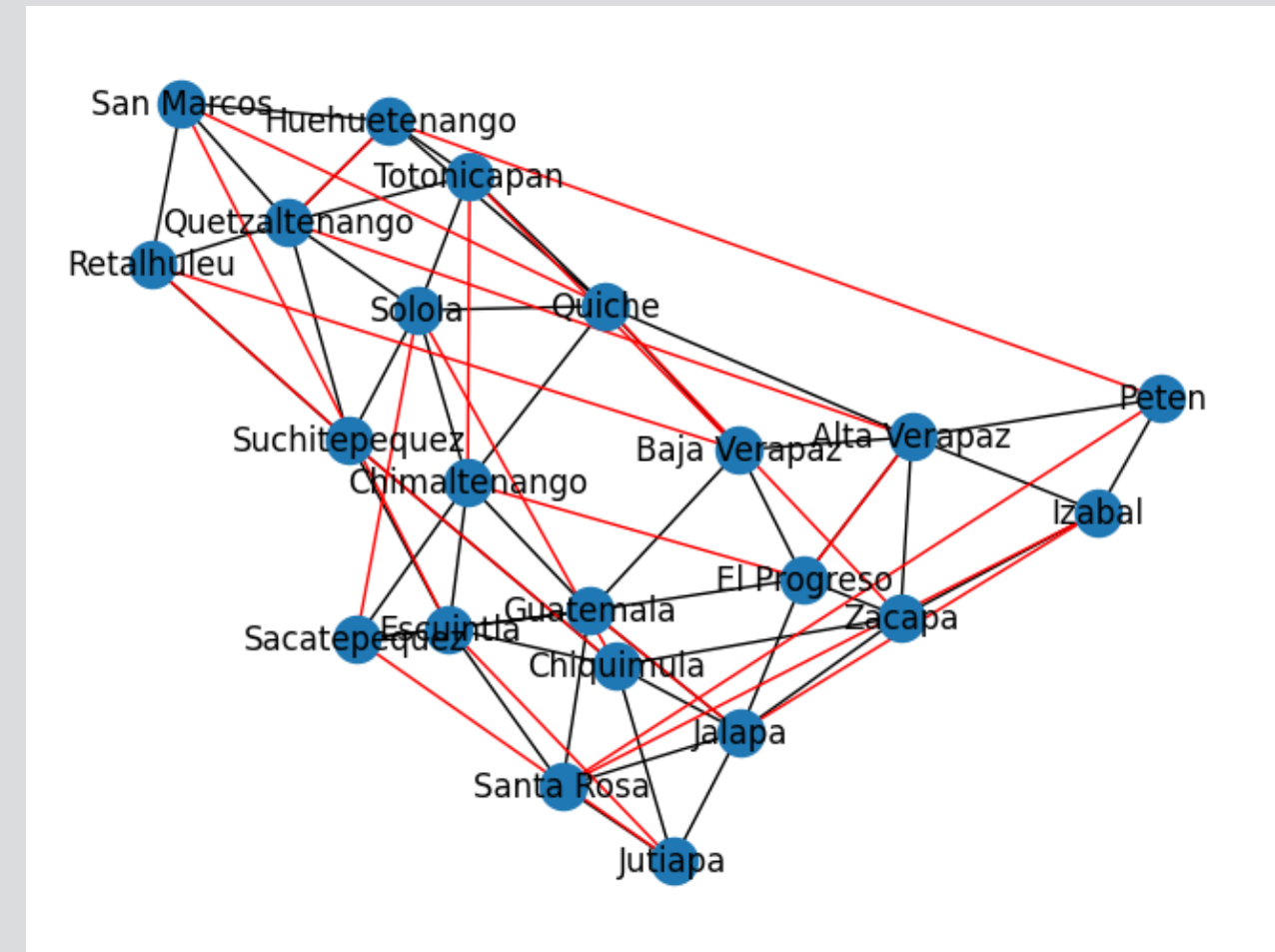
- Debido a que no todos los nodos se conectan entre sí, el algoritmo no llega a una solución en específico.
- Distancia retornada: inf

RETALHULEU A PETÉN

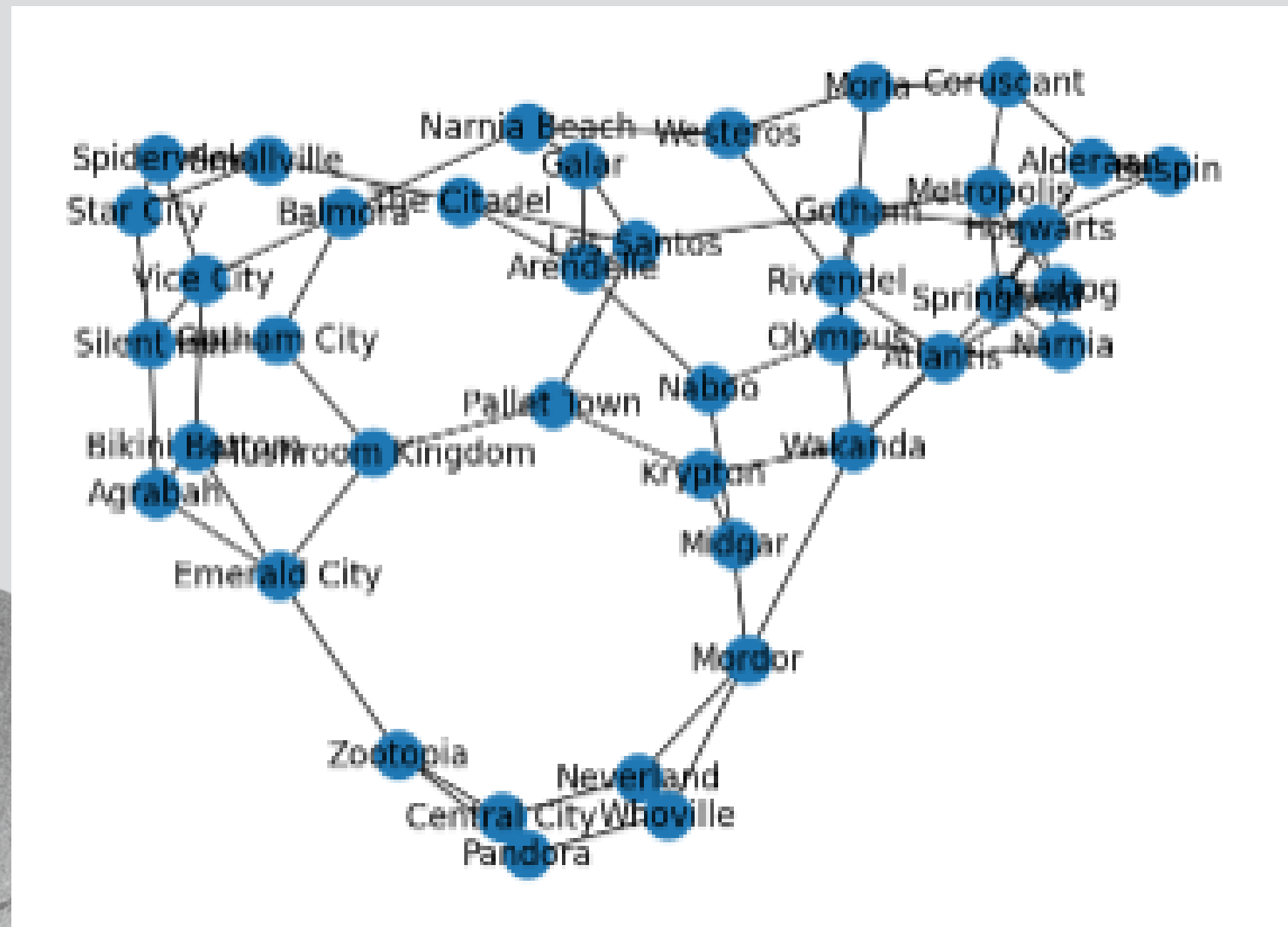
01. Dijkstra



02. Alg. Genético



ENCONTRANDO LA RUTA MÁS CORTA ENTRE 30 CIUDADES FICTICIAS

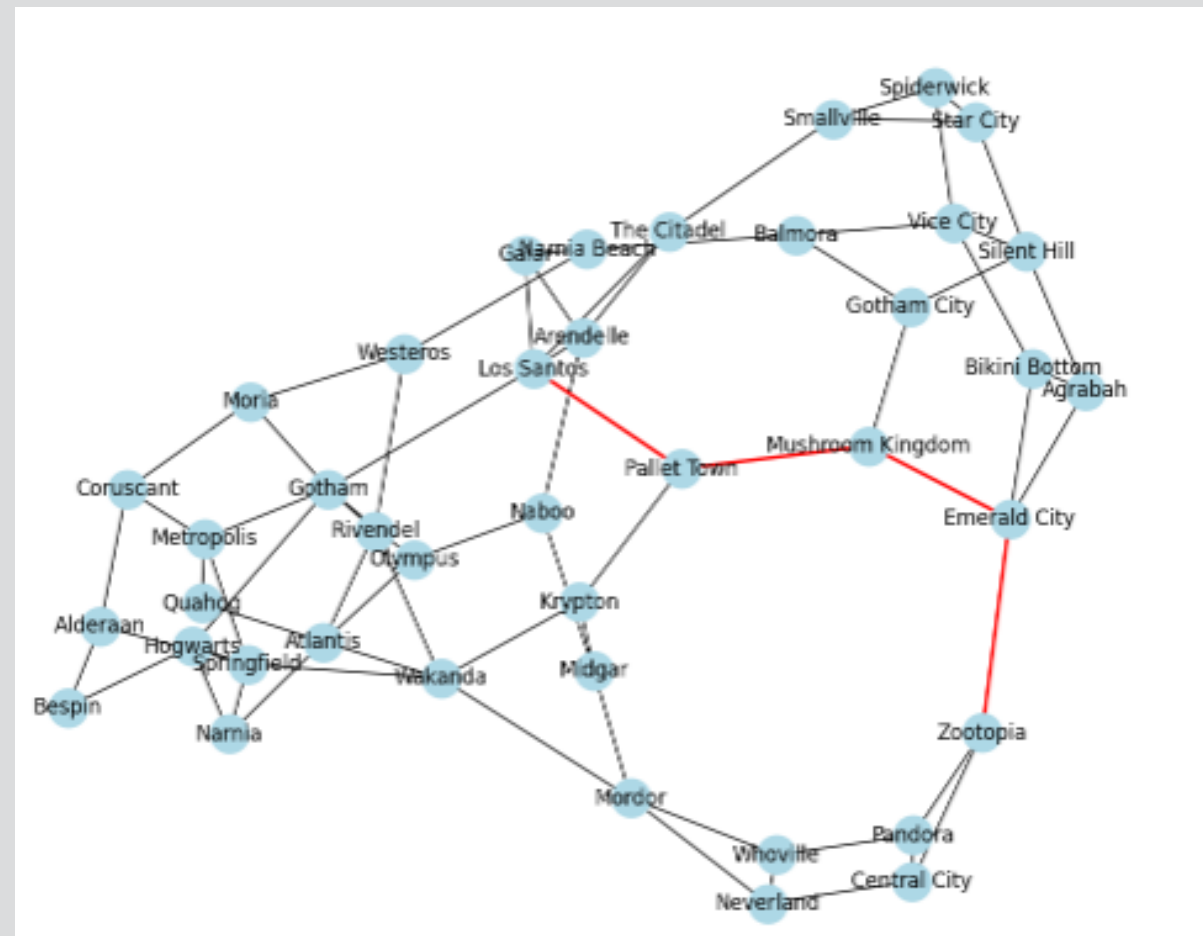


Se generó un segundo grafo en el que cada nodo es una ciudad ficticia y las aristas tienen un peso de la distancia entre sí;. Se buscará la ruta más corta entre dos nodos mediante los dos algoritmos y se compararán las soluciones obtenidas.

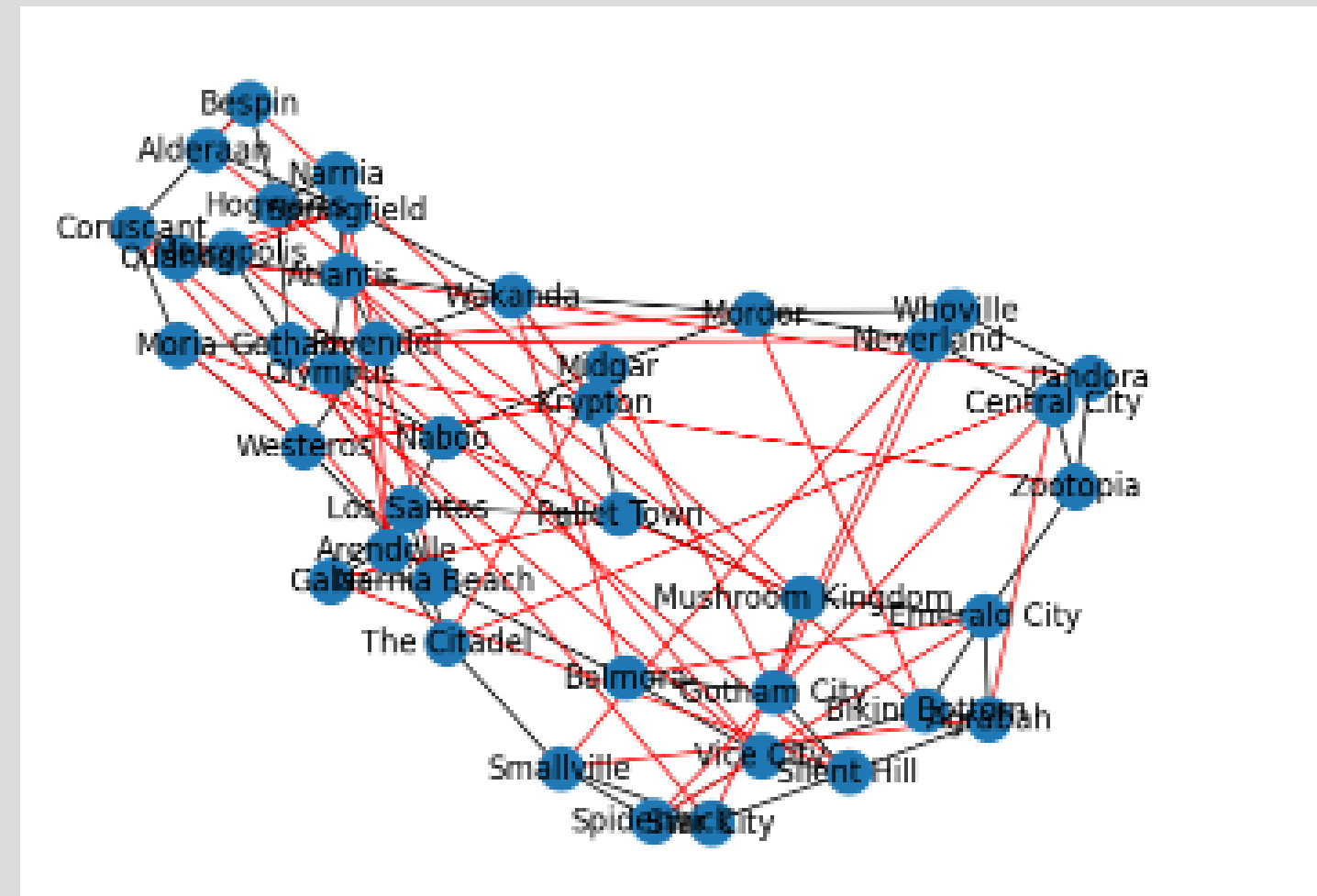
COMPARACIÓN DE SOLUCIONES

Ruta más corta de Zootopia a Los Santos

01. Dijkstra

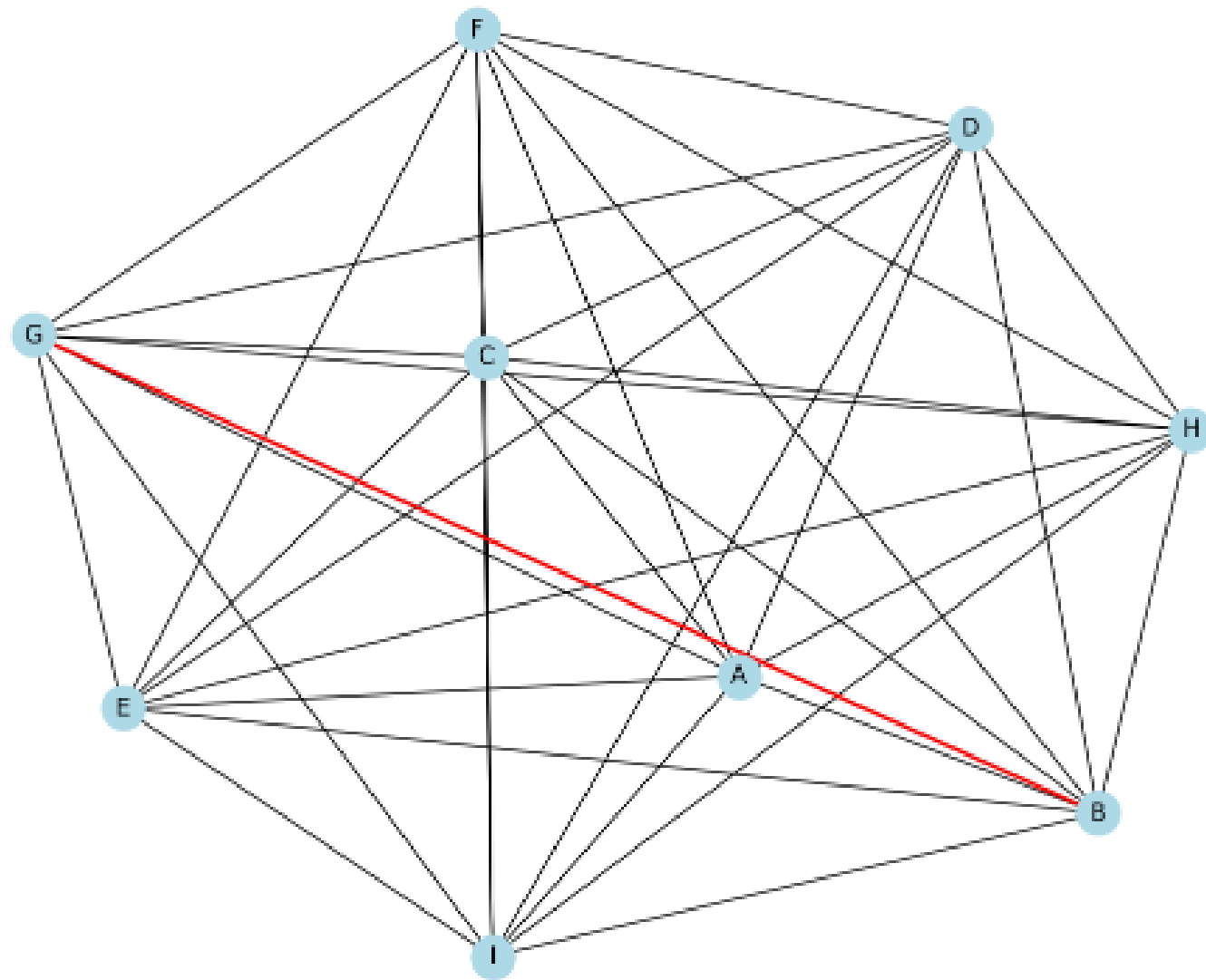


02. Alg. Genético



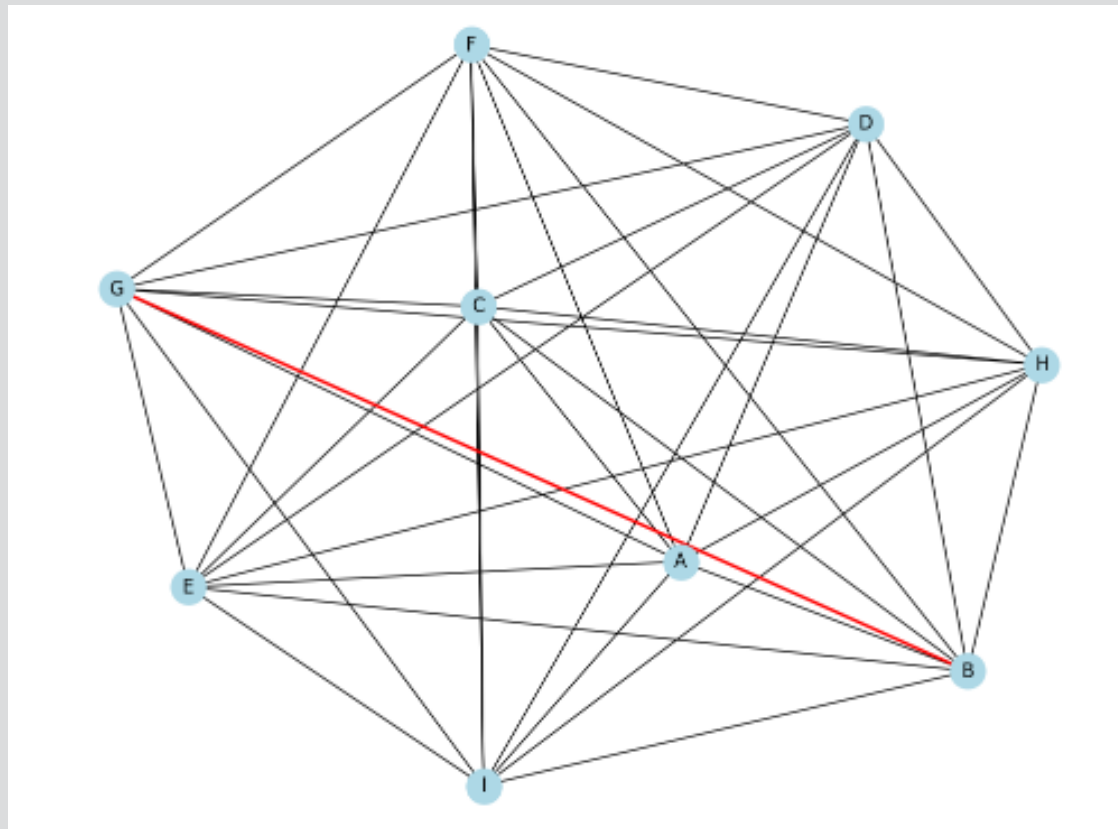
ENCONTRANDO LA RUTA MÁS CORTA EN UN GRAFO COMPLETAMENTE CONECTADO

Debido a que los algoritmos genéticos mejoran su rendimiento en grafo completamente conectado, se quiso estudiar el desempeño de los algoritmos con esta condición

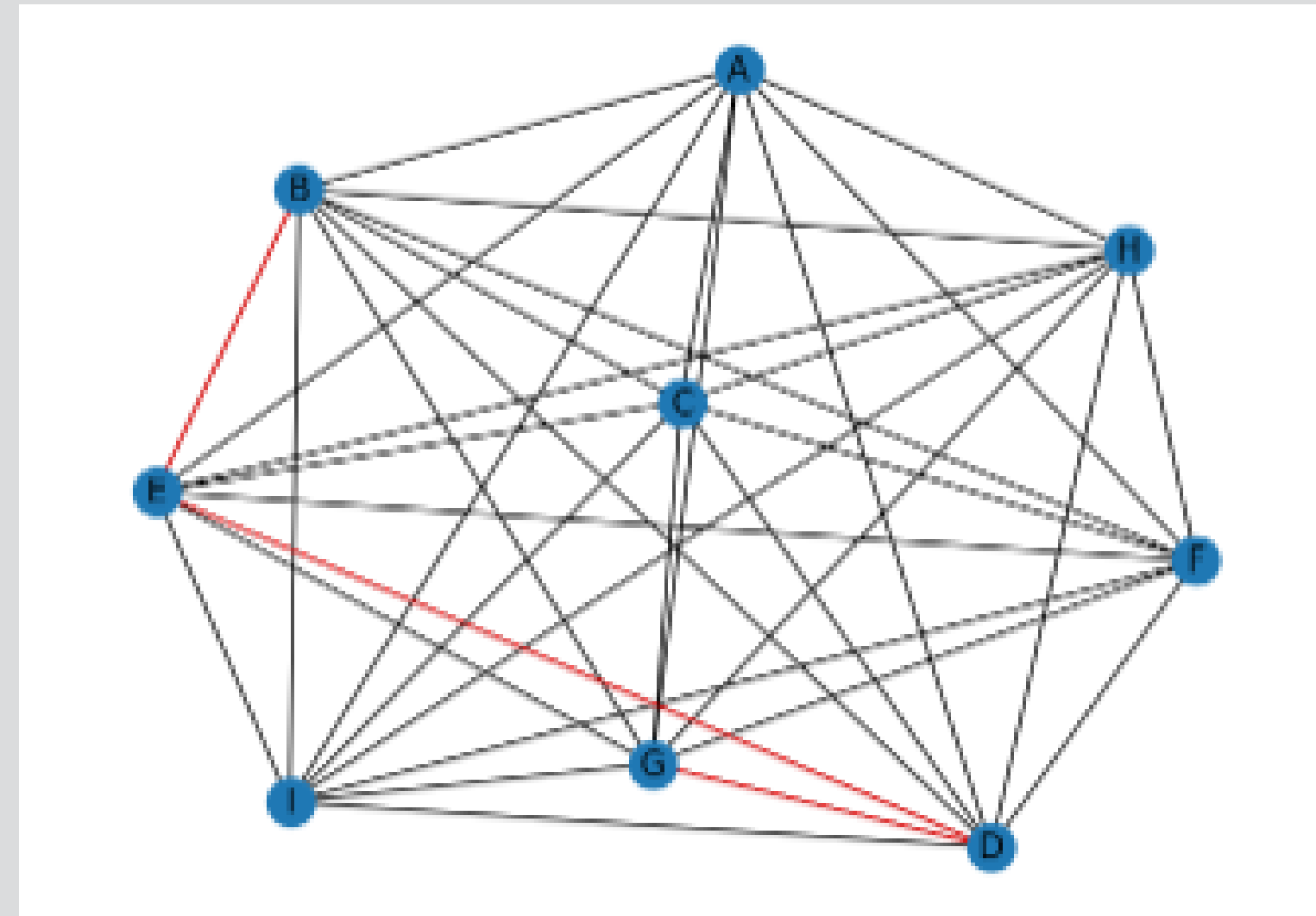


NODO B A NODO G

01. Dijkstra



02. Alg. Genético



OTRAS COMPARACIONES Y PERSPECTIVAS

Además de Dijkstra y algoritmos genéticos hay algunos otros algoritmos que se pueden usar para encontrar caminos entre dos puntos, existen algunos como DFS y BFS o AStar.

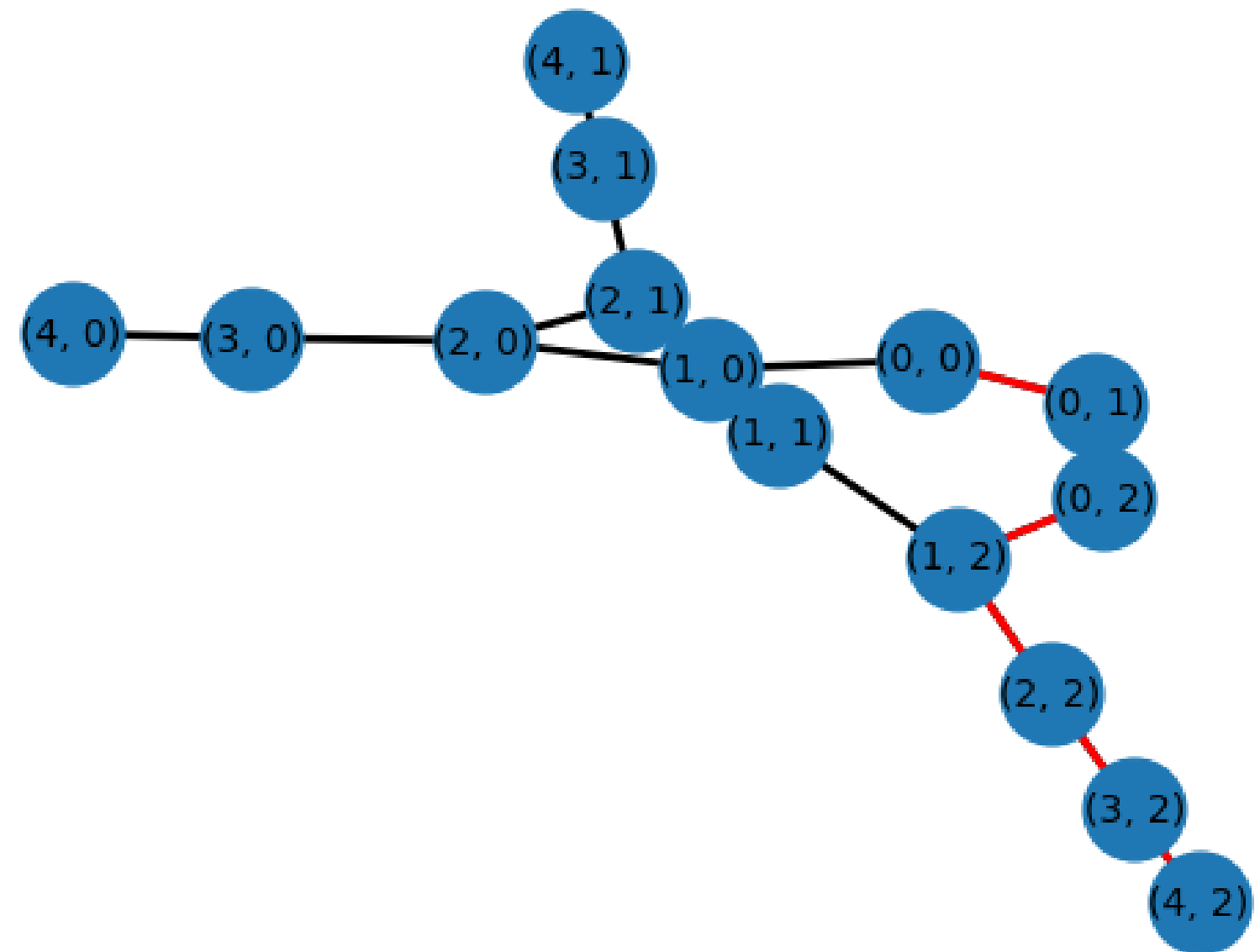
DFS tiene la desventaja que si la solución esta muy cercana pero en cierta dirección diferente a la que prioriza el algoritmo, tardará mucho en encontrarla o incluso podría no hacerlo. BFS revisa todos los nodos cercanos, por lo que siempre encontrará la ruta mas corta, sin embargo puede no ser muy eficiente ya que es como una red que se va expandiendo a través de los nodos hasta que la encuentra.

OTRAS COMPARACIONES Y PERSPECTIVAS

AStar (A^*) es algo similar a Dijkstra en cuanto a los criterios de para encontrar la ruta, solo que toma la distancia Manhattan lo cual en un problema en que la distancia entre todos los nodos es la misma puede ser más eficiente ya que en este caso, la distancia Manhattan se toma como la cantidad de nodos entre un nodo y otro, sin embargo si la distancia entre nodos no es igual entre todos podría no encontrar la mejor ruta.

ALGORITMO A*

DE (0,0) A (4,2)
SHORTEST PATH:
[(0, 0), (0, 1), (0, 2), (1, 2), (2, 2),
(3, 2), (4, 2)]



CONCLUSIONES

- Para los grafos cuyos nodos estén completamente conectados, el algoritmo ideal entre Dijkstra y Alg. Gen., es el de algoritmos genéticos, pues logra encontrar el valor óptimo en un menor tiempo.
- Para los grafos que no necesariamente todos sus nodos están conectados entre sí, el algoritmo de Dijkstra es el ideal, pues, sí logra encontrar la solución óptima en cuestión de segundos
- En grafos muy grandes el algoritmo Dijkstra puede no ser muy eficiente..
- Además de los algoritmos genéticos y Dijkstra. existen otros algoritmos que pueden ser útiles para resolver este problema. Entre estos, resalta el algoritmo A*, que resultó ser muy eficiente y dar resultados positivos cuando se implementó debido a que usa la guía heurística
- Los algoritmos de optimización combinatoria pueden no encontrar siempre una solución óptima

REFERENCIAS

- Amit's A* pages. (s/f). Stanford.edu.
<https://theory.stanford.edu/~amitp/GameProgramming/>
- Chiquimula — Zacapa, distancia (km, milla), ruta en el mapa, diferencia en el tiempo. (s/f). 2markers.com. <https://gt.2markers.com/461102-31099>
- Rasheed, S. A. (s/f). Solving shortest path problems using genetic algorithms. iasj.net. <https://www.iasj.net/iasj/download/1875f8c364732767>
- Russell, Stuart J. (2018). Artificial intelligence a modern approach. Norvig, Peter (4th ed.). Boston: Pearson. ISBN 978-0134610993. OCLC 1021874142.
- Shortest Path Algorithm — I — Data structures. (s/f). Inlibnet.Ac.In. <https://ebooks.inlibnet.ac.in/csp01/chapter/shortest-path-algorithm-i>
- Shortest path problems in the real world. (2018, octubre 18). Adam C. Conrad. <https://www.adamconrad.dev/blog/shortest-paths/>
- Song, J. [@JohnSong]. (2019, septiembre 15). A comparison of pathfinding algorithms. Youtube. <https://www.youtube.com/watch?v=GC-nBgi9r0U>